

Magdalena ŁADNIAK, Adam PIÓRKOWSKI, Mariusz MŁYNARCZUK  
Akademia Górniczo Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej

## KONSTRUKCJA SYSTEMÓW EKSPLORACJI DANYCH DLA OBRAZÓW RASTROWYCH <sup>1</sup>

**Streszczenie.** Poniższy artykuł porusza tematykę możliwości wydobywania wiedzy zawartej w obrazach rastrowych, magazynowanych w bazie danych. Zaproponowano autorskie podejście do możliwości konstrukcji systemu wspomagającego proces eksploracji takich danych. Omówione zostały aspekty konstrukcyjne oraz implementacyjne.

**Słowa kluczowe:** klasyfikacja danych, przetwarzanie obrazów, bazy danych, funkcje użytkownika

## STRUCTURE OF SYSTEMS FOR DATA EXPLORATION FOR RASTER IMAGES

**Summary.** The following article deals with possibilities for retrieving information from raster images stored in a database. An author's approach for the possibility of a support system construction for the process of exploration of such data is proposed. The following article describes construction and implementation aspects.

**Keywords:** data classification, image processing, database, user's functions

### 1. Wprowadzenie

Gwałtowny wzrost ilości informacji dostarczanych systemom bazodanowym w postaci danych dwuwymiarowych (jakimi są obrazy), zwiększa zapotrzebowanie na rozwój możliwości ich analizy. Zagadnienie eksploracji takich danych często opiera się na próbie sklasyfikowania ich według określonego wzorca lub zadanego parametru. Precyzyjne odwzorowania są nierzadko trudne, ze względu na różnorodność, zniekształcenie wzorca [1] bądź koniecz-

---

<sup>1</sup> Praca finansowana w ramach badań statutowych Katedry Geoinformatyki i Informatyki Stosowanej

ność analizy zmienności parametru. W takich przypadkach niezbędna jest dodatkowa interpretacja, zależna od konkretnie opracowywanego problemu. Automatyzacja procesu klasyfikacji obrazu jest zadaniem niełatwym, które niejednokrotnie wymaga dodatkowej wiedzy użytkownika. W niniejszym artykule przedstawiono możliwości konstrukcji systemu wspomagającego automatyzację wyszukiwania informacji zawartych w obrazach. Idea zaproponowanego rozwiązania ma na celu ułatwienie wydobywania wiedzy z wielu obrazów jednocześnie, bez konieczności tworzenia dodatkowego oprogramowania.

Odkrywanie wiedzy z bazy danych jest procesem złożonym, często reprezentowanym przez zestaw kroków [2, 3]:

- selekcja danych – wybór informacji dla analizy,
- transformacja danych – przygotowanie danych,
- eksploracja – ekstrakcja wiedzy,
- interpretacja wyników.

Zaproponowane rozwiązania, dotyczące możliwości wydobywania wiedzy zawartej w obrazach rastrowych magazynowanych w bazie danych, opierają się na analogicznym zestawie kroków. Niestety klasyczne podejście do eksploracji danych, wykorzystujące zewnętrzne narzędzia statystyczne, nie dopuszcza możliwości analizy danych zapisanych w postaci binarnej. Zaproponowane poniżej rozwiązania wykorzystują możliwości przetwarzania obrazów po stronie bazy danych, w celu wydobycia szukanej informacji, zwalniając z konieczności tworzenia zewnętrznego oprogramowania pośredniczącego.

W badaniach jako zestaw danych testowych wykorzystano grupę liczącą sto obrazów rastrowych, przedstawiających mikroskopową budowę skał węglanowych. Celem analizowanego poniżej przykładu jest klasyfikacja badanych danych według miejsca pobrania próbki i stopnia jej porowatości. Porowatość określana jest jako właściwość substancji stałej, polegająca na istnieniu w niej przestrzeni niewypełnionych, tj. porów. Materiał badawczy przed wykonaniem badania mikroskopowego poddano impregnacji klejem z dodatkiem barwnika. Uzyskano dzięki temu efekt wyraźnego zaznaczenia przestrzeni porowej, jej kształtu i wielkości. W ten sposób wszystkie istniejące w skale otwarte przestrzenie porowe zostały zabarwione na kolor niebieski [4], co było podstawą wyboru algorytmów segmentacji obrazów opisanych poniżej.

Tabela 1  
Pochodzenie i typ badanych skał

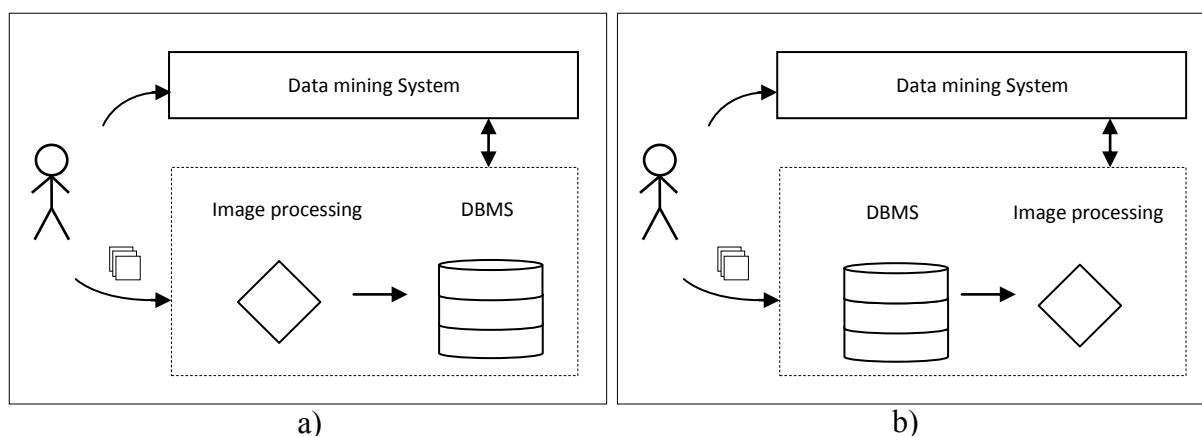
Pochodzenie	Typ
Buszewo	dolomit sparytowy
Baczyna	dolomit oolitowy
Kościan	dolomit drobnosparytowy
Santok	wapień oolitowy

## 2. Konstrukcja systemu eksploracji danych zawartych w obrazach

Zaproponowane podejście oparte jest na interakcji użytkownika z systemem bazodanowym (DBMS, ang. *Database Management System*) oraz narzędziem analitycznym. Wykorzystane zostały również możliwości implementacji algorytmów przetwarzania obrazów z zastosowaniem bazodanowych funkcji użytkownika (ang. *User-Defined Functions*). Możliwości konstrukcji omawianego systemu eksploracji danych uzależnione są od indywidualnych potrzeb analityka. Już na etapie projektu zalecane jest, by zdefiniować wymogi funkcjonalne, znać przedmiot analizy oraz sposób reprezentacji wyników. Wiedza ta staje się bardzo użyteczna w momencie szacowania wydajności systemu, co ma kluczowe znaczenie podczas wyboru architektury oraz sposobu implementacji.

### 2.1. Kontekst konstrukcji systemu

Analiza potrzeb ma znaczący wpływ na wybór konstrukcji rozwiązania. Jednym z czynników, który wpływa na kolejność przepływu informacji w systemie, jest parametryzacja poszukiwanych cech. Inna powinna być architektura systemu dla poszukiwania informacji, które wymagają parametryzacji w czasie rzeczywistym, a inna dla stałych elementów analizy. Rozróżnienie to ma wpływ na rozmiar magazynowanych wyników, ich trwałość oraz czasy wykonania rozwiązania. Poniższy rysunek prezentuje dwa przykładowe podejścia, obrazujące kolejność przetwarzania danych.



Rys. 1. Przetwarzanie obrazów przed (a) oraz po (b) żądaniu użytkownika  
Fig. 1. Image processing before (a) and after (b) user requests

Analiza obrazów oraz umieszczenie wyników w bazie danych przed żądaniem użytkownika (rys. 1a) są korzystne i możliwe do wykonania w przypadku, gdy wynik ma stały i nieparametryczny charakter. Przykładem może być analizowany proces wyliczenia współczynników porowatości dla zestawu danych. Problem taki nie wymaga ingerencji badającego w czasie rzeczywistym, możliwe zatem jest jego wykonanie w dowolnym czasie działania

systemu. Może to być moment umieszczania zdjęć w bazie danych lub praca w tle, gdy system jest nieobciążony. Wynik takich obliczeń, mający charakter ilościowy, opatrzony typem akceptowalnym przez popularne narzędzia statystyczne, nadaje się do dalszego uczestnictwa w procesie analizy.

Innym przypadkiem jest potrzeba wymagająca parametryzacji w czasie rzeczywistym. Jako elementarny przykład posłużyć może wyszukiwanie informacji o procentowej zawartości pikseli w dowolnie zdefiniowanych barwach. Wykorzystanie do tego celu funkcji użytkownika, gdzie różne progi binaryzacji mogą być ustalane i na bieżąco przekazywane jako jej parametr, znacząco ułatwia segmentację dużej ilości obrazów. Dla takiego scenariusza polecany jest schemat drugi (rys. 1b), kiedy wykonanie procedury przetwarzającej obraz następuje dopiero w momencie zdefiniowania zapytania do systemu. Efekty tej metody mogą zostać zapisane do bazy danych (i interpretowane w późniejszym czasie) bądź analizowane przez użytkownika na bieżąco.

## 2.2. Przechowywanie obrazów i informacji w nich zawartych w bazach danych

Pliki graficzne w bazie danych umieszczane są w polach typu BLOB (ang. *Binary Large Object*). Taki typ danych umożliwia magazynowanie danych binarnych w systemach zarządzania bazami danych, jako pojedynczy obiekt. Szczególnym przypadkiem danych binarnych są dane multimedialne [5, 6], także obrazy.

IMAGES	RESULTS
<pre> + id INTEGER, PRIMARY KEY + name VARCHAR + image BLOB + description VARCHAR + width INTEGER + height INTEGER </pre>	<pre> + id INTEGER, FOREIGN KEY + result_1 SELECTED TYPE ... + result_n SELECTED TYPE </pre>

Rys. 2. Przykładowe tabele w bazie danych

Fig. 2. Sample Database Tables

Dane wejściowe w postaci obrazów umieszczane są z użyciem zewnętrznej funkcji w bazie danych. Przykładowy zestaw tabel w bazie danych prezentuje rysunek 2. Tabela *Images* przechowuje informacje o zestawie danych wejściowych. Wyniki procedur mogą być zapisywane wg uznania użytkownika, jako dodatkowa kolumna tabeli *Images* lub osobno, na wzór tabeli *Results* połączonej kluczem obcym *id*. Ich rozmiar może być dowolny, często uzależniony od liczby wyliczeń (każdy wynik trafia do nowej kolumny). Alternatywnym rozwiązaniem jest wykorzystanie obiektowo-relacyjnych baz danych, które mogą enkapsulować. Przykładową formą, która dopuszcza możliwość przechowywania różnych typów

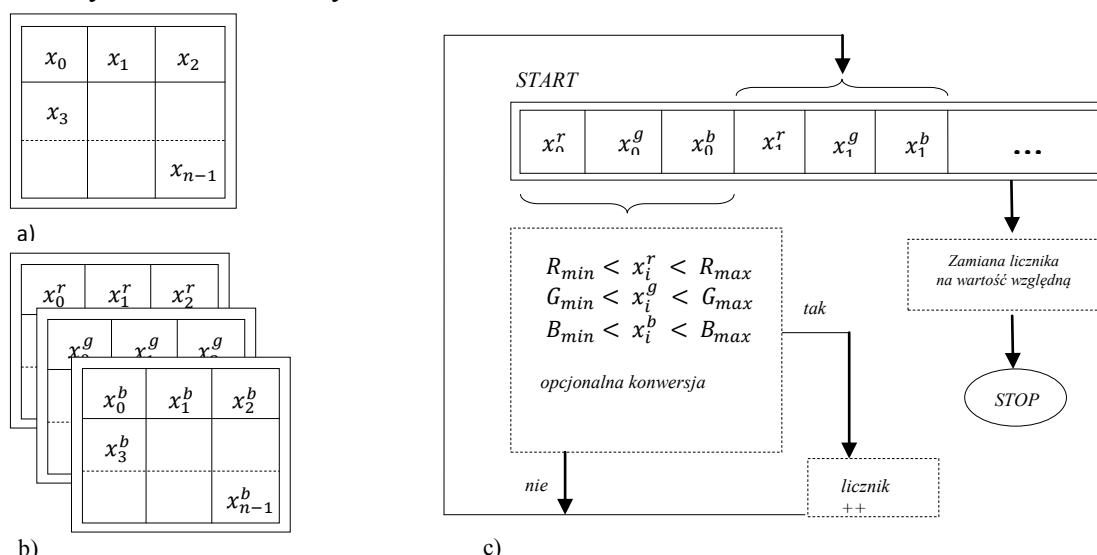
rozwiązań, jest XML (ang. *Extensible Markup Language*). Dla przejrzystości rozważań nie zostało to dokładniej opisane.

### 2.3. Funkcje użytkownika

Istnieją możliwości korzystania z zewnętrznych bibliotek, dedykowanych dla wybranych DBMS, których celem jest wykonywanie ściśle określonych działań w dziedzinie przetwarzania i analizy obrazu. Jednak zamierzeniem autorów nie jest ich wykorzystanie, a zaprezentowanie uniwersalnych możliwości zaimplementowania własnych algorytmów. Przetwarzanie obrazów po stronie bazy danych jest procesem czasochłonnym, dlatego w celu zwiększenia efektywności systemu, zaproponowano autorską metodę optymalizacji opisanego algorytmu segmentacji obrazu na podstawie analizy barwy.

#### 2.3.1. Algorytm wydobywania informacji z obrazu o procentowym udziale wybranego zabarwienia

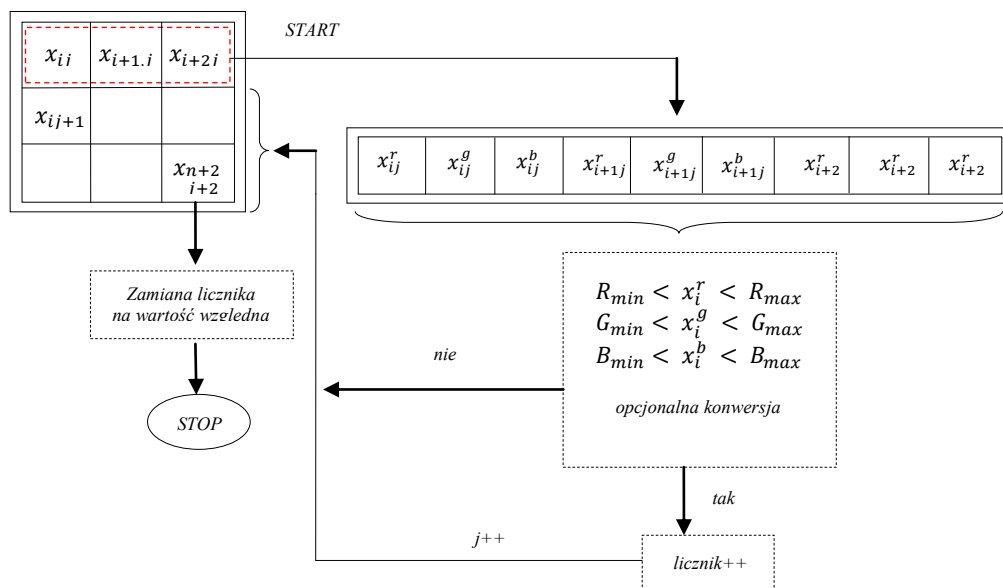
W komputerowej analizie obrazów celem przekształceń jest najczęściej otrzymanie poprawnie posegmentowanego obrazu binarnego. W sposób właściwy wyróżnione są na nim interesujące obiekty i tło. Aby otrzymać poprawny obraz binarny, należy uprzednio dokonać szeregu przekształceń obrazu szarego. W efekcie uzyskuje się kontrastowy obraz, na którym obiekty (np. pory) wyraźnie rozróżniają się od tła (np. ziarn) [7-8]. Przekształcenia stosowane w automatycznej analizie obrazów w zasadzie zdefiniowane są dla obrazów binarnych i szarych. Najczęstszym zaś sposobem analizy zdjęcia barwnego jest jego bezpośrednia zamiana wg określonych zasad na obraz szary.



Rys. 3. Graficzna interpretacja obrazu w formie: a) macierzowej, b) macierzowej z rozbiem na warstwy, c) algorytm odnajdywania procentowego udziału kompozycji barwnej

Fig. 3. Graphical interpretation of the image in the form of: a) matrix, b) matrix split into layers, c) algorithm for finding color composition

W celu detekcji przestrzeni porowej, z obrazów wydobywana jest informacja o wartościach poziomów szarości pikseli dla poszczególnych składowych barwy. W pierwszym przykładzie, w celu zobrazowania zagadnienia, skorzystano z podstawowego modelu przestrzeni barw, opisywanej współrzędnymi  $RGB$ . Wartości poszczególnych pikseli konwertowane są do postaci wektora o długości trzy razy dłuższej niż liczba pikseli obrazu wejściowego przed rozłożeniem na warstwy. Schemat działania algorytmu w wersji podstawowej został przedstawiony na rysunku 3. W każdej iteracji badany jest pojedynczy (kolejny) piksel pod kątem dopasowania do zadanych progów binaryzacji. Progi binaryzacji w analizowanym przykładzie dobrane zostały zarówno w sposób manualny, jak i z użyciem metody automatycznej.



Rys. 4. Zoptymalizowany algorytm odnajdowania procentowego udziału kompozycji barwnej  
Fig. 4. Optimized algorithm for finding color composition

Rysunek 4 przedstawia ten sam algorytm, z optymalizacją czasu wykonania (minimalizującą). Głównym krokiem optymalizacyjnym jest wydobywanie wartości poszczególnych pikseli nie z całego wektora, jakim jest obraz, a jego fragmentu, np. wiersza.

Powodem jest fakt, iż metoda służąca do pobierania podciągu znaków w niektórych systemach zarządzania bazami danych trwa stosunkowo długo. Czas ten jest proporcjonalnie zależny od ciągu znaków, z którego wybierany jest iteracyjnie podzbiór. Z każdą iteracją w algorytmie bez optymalizacji czytany jest cały wektor wejściowy. W takim podejściu każdorazowe pobranie jednego piksela wiąże się z bardzo dużym narzutem czasowym. Wybieranie pikseli z mniejszych wektorów (jak zaproponowano) powoduje szybsze działanie systemu, co potwierdzono zaprezentowanymi w dalszej części przykładami.

Dla każdego z zarejestrowanych obrazów przeprowadzono również badania automatycznego pozyskiwania progów binaryzacji na podstawie kroków:

- transformacja z przestrzeni barw RGB do przestrzeni YIQ – w celu wykrycia przestrzeni porowej wykorzystano składową luminancji Y modelu koloru YIQ. Transformacja modelu barw RGB do YIQ jest przedstawiona poniższym równaniem:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & -0,311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

- automatyczny dobór progu binaryzacji metodą Otsu [9] wg algorytmu:
  - wyznacz wartość wariancji międzyklasowej dla każdej wartości progowej,
  - progiem optymalnym jest wartość, dla której ta wariancja osiąga wartość maksymalną.
 Ze względu na ograniczoną ilość miejsca, w niniejszej publikacji nie definiowano przekształceń oraz definicji funkcji tworzących matematyczny opis metody. Szczegółowo opracowaną metodę można znaleźć w pozycji źródłowej [9].

### 2.3.2. Przykłady implementacyjne funkcji użytkownika dla wybranych systemów zarządzania bazami danych

Badanie zostało przeprowadzone dla czterech systemów bazodanowych: PostgreSQL, MySQL, MSSQL, Oracle. Poniżej zaprezentowano przykładowy kod tworzenia tabeli, w której znajdują się podstawowe magazynowane dane:

```
CREATE TABLE Images
(
  id INT PRIMARY KEY NOT NULL,
  name VARCHAR(50) NOT NULL,
  image BYTEA NOT NULL,
  height INT NOT NULL,
  width INT NOT NULL,
  description VARCHAR(250)
);
```

Każdy z przetestowanych systemów baz danych ma własną reprezentację pola BLOB, choć sens ich pozostaje niezmienny. Różne są również możliwości dostępu do wartości poszczególnych pikseli. Poniższa tabela przedstawia najważniejsze podstawowe różnice implementacyjne [10, 11] dla przykładu wydobywania wartości piksela z warstwy czerwonej.

Tabela 2  
Podstawowe różnice implementacyjne metody pobrania wartości pikseli

	sposób wybierania bajtów	reprezentacja pola BLOB
<b>MS SQL</b>	SET @r=CAST(SUBSTRING(@image,@i,1) AS INT);	VARBINARY
<b>MySQL</b>	SELECT conv(hex(substring(image,i,1)),16,10) INTO r;	LONGBLOB
<b>Oracle</b>	r := to_number(dbms_lob.substr(image,1,i), 'xx');	BLOB
<b>PostgreSQL</b>	r = get_byte(substring(image from i for 1), 0);	BYTEA

Poniżej znajdują się przykładowe implementacje składowych algorytmu wydobywania informacji z obrazu o procentowym udziale wybranego zabarwienia w języku `plpgsql`:

a) próg binaryzacji, dobrany w sposób empiryczny dla przestrzeni RGB – bez optymalizacji

```
CREATE FUNCTION porosity(image BYTEA)
RETURNS FLOAT AS $$
DECLARE
  [...]
BEGIN
  [...]
  WHILE i <= length LOOP
    r = get_byte(substring(image from i for 1), 0);
    g = get_byte(substring(image from i+1 for 1), 0);
    b = get_byte(substring(image from i+2 for 1), 0);
    IF (r>16 AND r<165 AND g>110 AND g<239 AND b<256 AND b>148) THEN
      counter = counter + 1;
    END IF;
    i = i + 3;
  END LOOP;
  result = (counter / (length / 3)) * 100;
  RETURN result;
END;
```

b) próg binaryzacji, dobrany w sposób empiryczny dla przestrzeni RGB – z optymalizacją

```
CREATE FUNCTION porosity_optimized(image BYTEA, width INT, height INT)
RETURNS FLOAT AS $$
DECLARE
  [...]
BEGIN
  [...]
  fragment_length = width * 3;
  WHILE j <= height LOOP
    image_fragment = substring(image from (1 + (j-1)*fragment_length) for
    fragment_length);
    WHILE i <= fragment_length LOOP
      r = get_byte(substring(image_fragment from i for 1), 0);
      g = get_byte(substring(image_fragment from i+1 for 1), 0);
      b = get_byte(substring(image_fragment from i+2 for 1), 0);
      IF (r>16 AND r<130 AND g>110 AND g<239 AND b<256 AND b>148) THEN
        counter = counter + 1;
      END IF;
      i = i + 3;
    END LOOP;
    j = j + 1;
    i = 1;
  END LOOP;
  [...]
END;
```

c) wyznaczenie luminancji poprzez transformację z przestrzeni RGB

```
[...]
image_row = substring(image_row from (1 + (j-1)*row_length) for row_length);
WHILE i <= fragment_length LOOP
  r = get_byte(substring(image_row from i for 1), 0);
  g = get_byte(substring(image_row from i+1 for 1), 0);
  b = get_byte(substring(image_row from i+2 for 1), 0);

  lamination_pixel = 0.299 * float4(r) + 0.587 * float4(g) + 0.114 *
  float4(b);
```



```
        lumination_image = set_byte(lumination_image, k, cast(lumination_pixel
AS integer));
        [...]

END LOOP;
```

#### d) automatyczny dobór progu binaryzacji metodą Otsu dla obrazu szarego

```
CREATE FUNCTION find_threshold_using_Otsu(image BYTEA, width INT, height INT)
RETURNS INT AS $$
DECLARE
[...]
BEGIN
[...]
    histogram = array_fill(0,ARRAY[256]);
    [...]
    WHILE j <= height LOOP
        image_row = substring(image from (1 + (j-1)*row_length) for
row_length);
        WHILE i <= row_length LOOP
            pixel = get_byte(substring(image_row from i for 1), 0);
            histogram[pixel] = histogram[pixel] + 1;
            sum = sum + pixel;
            [...]
        END LOOP;
        [...]
    END LOOP;

    FOR k IN 0..255 LOOP -- dla wszystkich poziomów szarości

        weightBackground = weightBackground + histogram[k+1];
        CONTINUE WHEN weightBackground = 0;
        weightForeground = length - weightBackground;
        EXIT WHEN weightForeground = 0;

        sumB = sumB + float4(k * histogram[k+1]);
        meanBackground = sumB / weightBackground;
        meanForeground = (sum - sumB) / weightForeground;

        varianceBetween = float4(weightBackground) * float4(weightForeground) *
(meanBackground - meanForeground) * (meanBackground - meanForeground);

        IF varianceBetween > varianceMax THEN
            varianceMax = varianceBetween;
            threshold = k;
        END IF;
    END LOOP;
    RETURN threshold;
END;
```

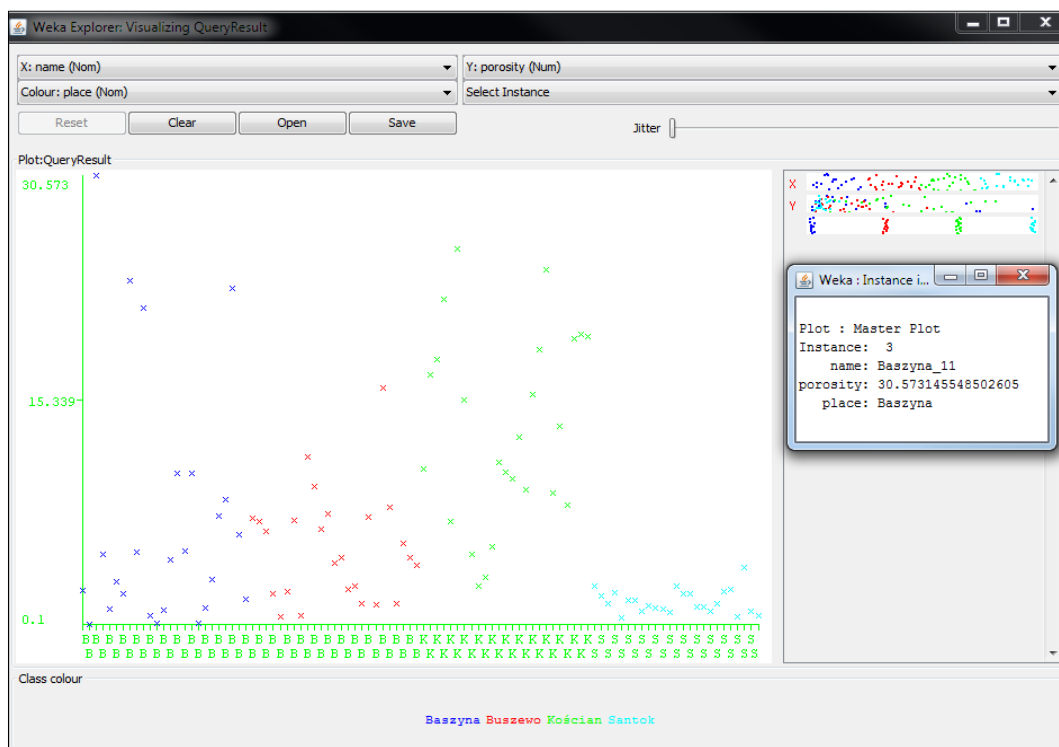
## 2.4. Powiązanie z narzędziem statystycznym

Istnieje wiele narzędzi dla całego procesu przekształcania danych w użyteczną wiedzę, począwszy od wprowadzenia danych, aż do utworzenia wynikowego raportu [12]. Dla zobrazowania przykładu wybrano program Weka (ang. *Waikato Environment for Knowledge Analysis*). Jest to program rozpowszechniany jako oprogramowanie typu *open source* na licencji GNU (ang. *General Public License*) [13]. W badaniach wykorzystano stabilną wersję pro-

gramu Weka 3.6.8. Skorzystano z zakładki ‘*Explorer*’, która zawiera panel umożliwiający analizę danych oraz bezpośrednią komunikację z systemami bazy danych.

Celem pracy było opracowanie konstrukcji systemu bazodanowego do wspomaganie pomiarów struktur skalnych metodami analizy obrazu. Metody te są w ostatnich latach z powodzeniem wprowadzane do badań związanych z geologią [14], górnictwem [15] oraz mechaniką skał [16]. Wymagają one jednak każdorazowej modyfikacji algorytmu oraz podawania ścieżek dostępu do plików z obrazami skał. Proponowany system pozwala na przeniesienie obliczeń na platformy bazodanowe, czym wychodzi naprzeciw oczekiwaniom użytkowników zajmujących się analizą obrazu [17]. Do prowadzonych w niniejszej pracy analiz wykorzystano próbki czterech skał węglanowych, dla których wyznaczono współczynnik porowatości oraz pogrupowano wyniki w zależności od miejsca pobrania analizowanej skały. Do wykonania tego zadania użyto wersji scenariusza 2 z podpunktu 2.1, czyli analiza obrazu po stronie bazy danych wykonywana jest po wysłaniu żądania przez użytkownika, co obrazuje poniższy przykład:

```
SELECT name, porosity_optimized(Image, Width, Height) AS Porosity,
       description AS place
FROM Images ORDER BY name;
```



Rys. 5. Weka Explorer – wizualizacja wyników zapytania  
Fig. 5. Weka Explorer – visualizing QueryResult

Efektom wykonania funkcji jest wczytanie wyników do systemu. Gdy format danych wyników jest akceptowalny przez system Weka, można przystąpić do dalszych analiz.

Rysunek 5 przedstawia graficzną interpretację rozwiązania problemu. Na osi X znajdują się nazwy miejsc pobrania próbki, natomiast oś Y przedstawia wartości wyliczonego współczynnika. Na tej podstawie można wywnioskować, że najmniejsza ilość wolnej przestrzeni w skałach znajduje się w skałach pochodzących z miejscowości Santok (na rysunku 5 jest to zestaw danych wysuniętych najbardziej na prawo).

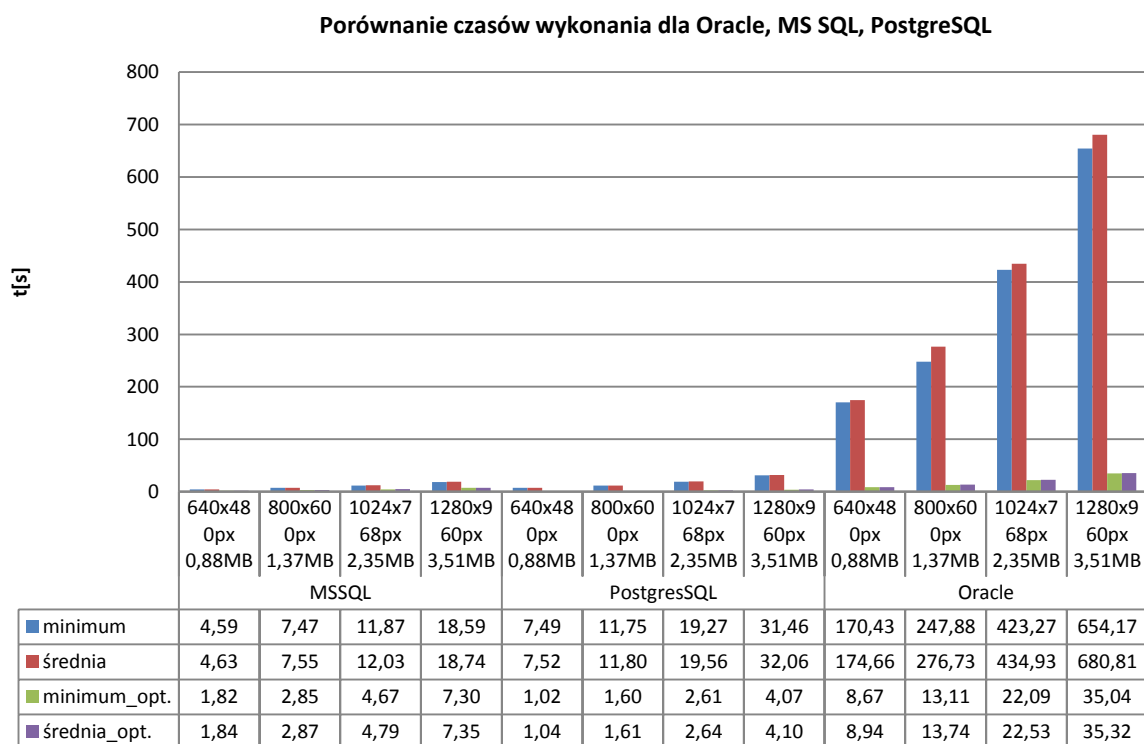
### 3. Testy wydajnościowe

Poniżej zaprezentowano zestawienie czasów wykonania przykładowych procedur wydobywających informację z obrazów metodami przetwarzania obrazów dla czterech systemów bazodanowych.

Testy przeprowadzone zostały na komputerze klasy PC o parametrach:

- dysk twardy: Seagate 2,5TB SATA, średni czas dostępu 14,9 ms,
- procesor: Intel® Celeron® CPU E3200 2x 2,40 GHz,
- pamięć: DDR II RAM 4,00 GB.

Każde obliczenie zostało wykonane z dziesięciokrotnym powtórzeniem. Do wiadomości podano wartości wyrażone w sekundach: minimalną, uśrednioną oraz odchylenie standardowe.



Rys. 6. Zestawienie czasów wykonania funkcji analizy dla wybranych systemów bazodanowych  
Fig. 6. Function execution time for selected database systems

Zestawienia nie uwzględniają czasów wykonania dla systemu MySQL, którego uśrednione wartości dla wersji zarówno zoptymalizowanej, jak i niezoptymalizowanej są zdecydowanie większe. Zaobserwowane różnice sięgają nawet stukrotnej wartości zmierzonego czasu dla omówionych wcześniej trzech systemów. Jest to powodem pominięcia rzeczzonego w dalszych rozważaniach.

Tabela 3

Zestawienie odchyłeń standardowych dla czasów wykonania funkcji analizy,  
dla wybranych systemów bazodanowych

	optymalizacja	640×480 px 0,88 MB	800×600 px 1,37 MB	1024×768 px 2,35 MB	1280×960 px 3,51 MB
Oracle	nie	3,13456	17,96124	6,39961	18,13411
	tak	0,34121	0,56561	0,26885	0,23949
MSSQL	nie	0,04905	0,07324	0,17518	0,12538
	tak	0,01360	0,01194	0,09500	0,02346
PostgresSQL	nie	0,03243	0,03148	0,55189	0,47088
	tak	0,01119	0,00888	0,01818	0,02369

#### 4. Podsumowanie

Przeprowadzone badania pokazują, iż możliwe jest skonstruowanie użytecznego systemu wspomagającego eksplorację danych zawartych w obrazach rastrowych bez użycia dodatkowego oprogramowania. Głównym celem badań było znalezienie i zaprezentowanie możliwości konstrukcji systemu do eksploracji wiedzy zawartej w obrazach magazynowanych w bazie danych, w postaci binarnej. Zademonstrowano możliwości rozwiązania tego problemu poprzez dekompozycję danych. Zaproponowano różne podejścia projektowe i implementacyjne, dostarczając przykładowych rozwiązań. Zaprezentowane testy wydajnościowe przedstawiają średnie potencjalne czasy wykonywania funkcji wchodzących w skład narzędzi systemu. W przyszłych badaniach podjęte zostaną dalsze próby poszukiwań optymalnych i użytecznych rozwiązań. Autorzy chcieliby wykonać bardziej dogłębne analizy zagadnienia optymalizacji odczytu i zapisu danych binarnych. Prowadziłoby to do zwiększenia wydajności czasowej, by realna stała się możliwość wykorzystywania takiego systemu dla bardzo dużych zbiorów danych, w czasach bliskich rzeczywistym. W planach rozwoju tematyki są opracowanie i ocena wydajności szerszej grupy algorytmów umożliwiających dokonanie uniwersalnych i niezbędnych do dalszych analiz przekształceń (np. wykorzystując metody morfologii matematycznej) obrazów, w celu zwiększenia zasięgu zastosowań. Kierunki dalszych badań będą zmierzały ku pełnemu wykorzystaniu w mineralogii zagadnień morfologii matematycznej [18-20].

**BIBLIOGRAFIA**

1. Zakrzewicz M.: Data Mining i odkrywanie wiedzy w bazach danych. Materiały konferencyjne III Konferencji Polskiej Grupy Użytkowników Systemu Oracle, Zakopane 1997.
2. Świtoński A., Zghidi H.: Wyszukanie wzorców w bazie danych obrazów dwuwymiarowych, [w:] Kozielski S., Małyśiak B., Kasprowski P., Mrozek D. (red.): Bazy Danych: Rozwój metod i technologii. WKŁ, Warszawa 2008.
3. Zaiane O. R.: Resource and knowledge discovery from the Internet and multimedia repositories. Simon Fraser University, British Columbia, Canada 1999.
4. Młynarczuk M.: Stereologiczny opis skał okruchowych przy użyciu metod analizy obrazu. Polskie Towarzystwo Mineralogiczne – Prace Specjalne, Vol. 27, 2005.
5. Ruoming J., Agrawal G.: A middleware for developing parallel data mining applications. Proc. SDM'2001: First SIAM International Conference on Data Mining, Chicago, USA 2001.
6. Stones R., Matthew N.: Bazy danych i MySQL. Helion, Gliwice 2003.
7. Młynarczuk M.: Możliwości wykorzystania analizy obrazu i morfologii matematycznej do analizy stereologicznej struktur skalnych. Archives of Mining Sciences, Vol. 49, 2004, s. 117÷140.
8. Młynarczuk M.: Application of Image Analysis and Mathematical Morphology in Petrography. Electrical Review No. 3, 2005, s. 20÷25.
9. Otsu N.: A Treshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man and Cybernetics, 9, 1, 1979, s. 62÷69.
10. Stones R, Matthew N.: Bazy danych i PostgreSQL. Helion, Gliwice 2003.
11. PostgreSQL Global Development Group, PostgreSQL 9.1.2 Documentation, <http://www.postgresql.org/docs/9.1/static>.
12. Oracle Corporation, MySQL 5.0 Reference, <http://dev.mysql.com/doc/refman/5.0/en>.
13. Chuchro M, Piórkowski A.: Wykorzystanie metod i narzędzi eksploracji danych do analizy zmienności natężenia dopływu do komunalnych oczyszczalni ścieków. Studia Informatica, Vol. 31, No. 2B (90), Gliwice 2010, s. 347÷358.
14. Młynarczuk M.: Some Remarks on the Application of Image Analysis and Image Processing for the Description of the Geometrical Structures of Rock. Physicochemical Problems of Mineral Processing, 33, 1999, s. 107÷116.
15. Wierzbicki M., Młynarczuk M.: Structural aspects of gas and dolomite outburst in Rudna copper mine, Poland. International Journal of Rock Mechanics & Mining Sciences, Vol. 57, 2012, s. 113÷118.

16. Młynarczuk M.: Description and classification of rock surfaces by means of laser profilometry and mathematical morphology. *International Journal of Rock Mechanics and Mining Sciences*, Vol. 47, Issue 1, 2010, s. 138÷149.
17. Lay B. J.: Image processing software – multiple solutions for a single problem. *Proceedings of the 9th European Congress on Stereology and Image Analysis.*, Zakopane 2005, s. 55÷70.
18. Matheron G.: *Random Sets and Integral Geometry*. Wiley, New York 1975.
19. Serra J.: *Image Analysis and Mathematical Morphology*. Academic Press, New York 1982.
20. Serra J., Młynarczuk M.: Morphological merging of multidimensional data. *Proc. Sixth. Intern. Conf. Stereology and Image Analysis in Materials Science. STERMAT 2000. Cracow 2000*, s. 385÷390.

Wpłynęło do Redakcji 8 stycznia 2013 r.

### **Abstract**

The following article discusses the possibilities of extracting information contained in raster images stored in a database. The author proposes his approach to the assistance of the process of such data with the use of support system. Construction and implementation aspects are discussed. The main operation is processing the input data using functions of the user for the analysis of the system and the proper preparation of the database with the results to work with popular data retrieval systems.

### **Adresy**

Magdalena ŁADNIAK: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, al. Mickiewicza 30, 30-059 Kraków, Polska, mladniak@geol.agh.edu.pl.

Adam PIÓRKOWSKI: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, al. Mickiewicza 30, 30-059 Kraków, Polska, pioro@agh.edu.pl.

Mariusz MŁYNARCZUK: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, al. Mickiewicza 30, 30-059 Kraków, Polska, mlynar@agh.edu.pl.